Fog Computing Integration for Real-Time IoT Data Processing

Zahraa kadhim Alitbi[1], seyed Amin Hosseini Seno [2]
[1]*College of Education for Pure Sciences, University of Wasit, Kut, Iraq*
[2]*Computer Engineering Department, Engineering Faculty, Ferdowsi University of Mashhad (FUM), Mashhad, IRAN*
z.ali@uowasit.edu.iq, hosseini@um.ac.ir

Keywords: Fog Computing, Internet of Things (IoT), Real-Time Data Processing, Task Offloading, Energy Efficiency, Edge Computing, Security, Quality of Service (QoS).

Abstract: The rapid expansion of the Internet of Things (IoT) has created massive streams of real-time data that require processing near their sources to ensure timely and efficient responses. Traditional cloud-centric architectures struggle to meet these demands, leading to significant latency, energy overhead, and security vulnerabilities. Fog computing, by extending computational and storage capabilities toward the network edge, offers a promising solution to these limitations. This study systematically analyses recent advancements in fog-enabled IoT data processing, consolidating performance results from diverse approaches into a unified comparative framework. The proposed model balances latency, energy consumption, and operational costs, demonstrating performance gains of up to 95% in latency reduction, 65% in energy savings, and notable improvements in system security. Through detailed comparative analysis and graphical evaluation, the findings reveal that multi-layer fog architectures, when combined with adaptive scheduling and energy-aware service placement, can significantly enhance quality of service (QoS) while optimising resource utilisation. These insights provide practical guidance for designing sustainable, secure, and high-performance IoT ecosystems.

1 INTRODUCTION

The rapid proliferation of Internet of Things (IoT) devices has transformed modern digital ecosystems, enabling a wide range of applications that require ultra-low latency, high reliability, and continuous connectivity—examples include autonomous vehicles, remote healthcare monitoring, industrial automation, and smart city services [1]. Traditional cloud-centric computing architectures face significant challenges in meeting these requirements, as data must traverse long network paths to distant data centres, leading to high latency, bandwidth inefficiencies, and potential network congestion [2]. These limitations are particularly critical in Industrial IoT (IIoT) environments, where real-time decision-making is essential to ensure operational safety, process optimisation, and service continuity [3]. Furthermore, transmitting all raw data to the cloud increases operational costs, energy consumption, and carbon footprint, while also introducing privacy and security concerns [4]. Fog computing has emerged as a promising paradigm to address these challenges by extending computation, storage, and analytics capabilities toward the network edge [5]. Fog nodes—such as gateways, access points, or micro data centres—can perform partial processing, filtering, and caching close to the data source, forwarding only the necessary information to the cloud [6]. This architecture significantly reduces latency, optimises bandwidth utilisation, and enables more energy-efficient and context-aware processing [5, 6].

While numerous studies have explored fog computing for IoT systems, the majority focus on individual performance metrics, such as latency reduction or energy efficiency, without offering a unified framework that simultaneously addresses latency, energy consumption, operational cost, and security. This lack of integrated analysis makes it difficult for system designers to make well-informed trade-offs that satisfy the diverse requirements of real-time IoT applications. To address this gap, this paper (1) systematically reviews state-of-the-art fog computing integration approaches for real-time IoT data processing; (2) consolidates and compares quantitative performance metrics from at least eighteen peer-reviewed studies; (3) proposes a unified mathematical cost model to balance latency, energy, and operational cost in offloading decisions; and (4) identifies open challenges and outlines future research directions with a focus on

sustainability, security, and adaptive architectures.

Through this comprehensive and comparative analysis, the paper provides practical guidance for researchers and practitioners seeking to design secure, sustainable, and high-performance fog-enabled IoT systems. Researchers have explored numerous fog-enabled architectures and algorithms. Dynamic offloading schemes decide which tasks to process locally, at fog nodes or in the cloud, achieving up to 95% latency reduction and 67% energy savings relative to cloud-only execution [6]. Hybrid edge–fog–cloud hierarchies for narrow-band IoT (NB-IoT) health monitoring reduce transmission delays by 59.9% and execution time by 38.5% [3]. Energy-aware service placement algorithms, such as LJAYA, cut energy consumption by 31% in fog networks [4], while green demand-aware techniques save up to 65% of energy without sacrificing delay performance [7]. To secure distributed fog environments, machine-learning-based authentication and intrusion detection schemes achieve 99.86% accuracy and 99.91% F1-score [8]. The next sections review these and other contributions, derive a composite cost function for offloading decisions, and analyse quantitative results across the literature.

2 Related Work

2.1 Dynamic task offloading and scheduling

Because fog nodes have limited resources, deciding where to execute a task is critical. Sensors et al. proposed a dynamic offloading threshold scheme incorporating dynamic task scheduling (DTS) and dynamic energy control (DEC) algorithms. Compared with benchmark cloud-only execution, the proposed method reduced latency by up to 95%, improved throughput by 71%, and cut energy consumption by 67% [2]. The authors highlight that fog computing mitigates high communication latency and network congestion by processing tasks close to devices [9]. Similarly, an energy-efficient IoT task scheduling framework (EEIoMT) classified tasks into critical, moderate and normal categories and computed weights based on energy and latency requirements. Simulations showed that EEIoMT decreased response time by 90%, network usage by 79%, cost by 80%, network latency by 65% and energy usage by 81% compared to cloud-based models [10].

Fuzzy-logic-enhanced scheduling algorithms have been proposed to handle uncertainty in task requirements. In DTA-FLE (Dynamic Task Allocation using Fuzzy Logic Enhanced approach), tasks are classified based on latency sensitivity using fuzzy sets, and a hierarchical scheduler assigns urgent tasks to fog nodes while deferring non-urgent tasks to cloud servers. The energy consumption of communication for a set of tasks on m resources is given by Eq. 1 [8]:

$$E_{\text{comm}} = \sum_{i=1}^{m} (P_{tx} + P_{rx})\, T_{\text{comm}}, \tag{1}$$

where P_{tx} and P_{rx} are transmission and reception power, respectively, and T_{comm} is communication time. The DTA-FLE scheduler minimises both energy and delay by reducing unnecessary task migrations; at 700 tasks the algorithm achieved a delay of 24 s, compared with 132 s for a cloud-only scheduler (DTA) and 35–50 min for other heuristics [11], corresponding to an 82% latency reduction relative to DTA. A modified Grey-Wolf optimisation (TS-GWO) for task scheduling further improved efficiency; experimental results showed makespan reductions of 46.15% and energy savings of 28.57% compared with other metaheuristic methods [5].

An energy-efficient time-and-cost (ETC) constraint scheduling algorithm based on an improved multi-objective differential evolution (I-MODE) has recently been proposed to optimise workflow applications in fog computing. Evaluations on synthetic and real-world workflows demonstrate that ETC lowers energy consumption by 14–24%, reduces execution time by 14–25%, and cuts monetary cost by 22–29% relative to baseline schedulers [12].

2.2 Hierarchical architectures

Fog architectures often adopt three layers: the device layer (sensors and actuators), the fog layer (near-edge nodes), and the cloud layer. Figure 1 illustrates this layered architecture. In the NB-IoT hybrid architecture for health monitoring [3], tasks such as data aggregation and preliminary analysis run on edge devices, while the fog layer performs protocol translation and local storage, and the cloud conducts long-term analytics. Simulations using CloudSim and iFogSim showed that the architecture reduced NB-IoT delay by 59.9% and execution time by 38.5% [3]. Energy-efficient service placement algorithms assign services to fog nodes based on resource availability and predicted workloads, reducing energy consumption by 31% [4]. In telehealth IoT systems, integrating fog nodes with a hybrid cloud platform and adaptive energy-saving strategies delivered a 2% energy reduction, demonstrating modest but non-negligible savings [13].

The hybrid fog–edge architecture for real-time health monitoring implements rule-based filtering and lightweight machine-learning models at edge nodes and uses fog nodes for time-critical tasks. Simulations showed 70% latency reduction, 30% energy efficiency improvement and 60% bandwidth savings relative to cloud-only deployments [14]. Such hybrid architectures also incorporate decision-tree and one-class support vector machines to classify abnormal signals and reduce network traffic. Green demand-aware schemes conserve energy by powering down idle fog nodes; a study reported up to 65% energy savings without increasing delay [15].

Another study introduced a power-aware fog-supported IoT healthcare network that optimises the deployment of heterogeneous gateways via swarm-intelligence algorithms; simulations demonstrate that the network can reduce power consumption by 33% [16]. Additionally, energy-consumption modelling for fog-enabled IoT communications found that energy usage is inversely proportional to the Maximum Transmission Unit (MTU) size—larger MTU values lead to lower energy consumption during data transmission.

2.3 Security and privacy

Fog computing's decentralised nature introduces new security challenges. A machine-learning-based authentication and intrusion-detection system combines elliptic curve cryptography with a stacked ensemble classifier. After secret-ID authentication, the system achieved 99.86% accuracy, 99.89% precision, 99.96% recall and 99.91% F1-score for detecting anomalies [8]. An adaptive encryption framework uses K-nearest neighbours to classify data sensitivity and applies hybrid ECC-AES encryption for sensitive data. Experimental evaluation measured encryption and decryption times (9.679–67.79 ms for sensitive data) and encryption throughput (0.826–14.75 MB s^{-1}); histogram analysis produced a Number of Pixels Change Rate (NPCR) of 99.349% and Unified Average Changing Intensity (UACI) of 33.079%, demonstrating strong resistance to statistical attacks.

A zero-trust fog-computing framework for healthcare integrates blockchain (BC) and software-defined networking (SDN) to enable continuous authentication. Using 50 IoT devices and 10 fog nodes in iFogSim, the authors reported a 40% improvement in intrusion detection rate, 30% enhancement in data integrity, 15.29% rise in task completion rate, and 39.66% reduction in average response time [16]. These gains illustrate that strong security mechanisms can coexist with low latency when implemented at the fog layer.

3 Unified Offloading Cost Model

Effective resource allocation in fog-enabled IoT systems requires a careful balance between latency, energy consumption, operational cost, and, in some cases, security considerations. These factors collectively influence the decision of where a given computational task should be executed—whether on the IoT device itself, a nearby fog node, or a remote cloud server.

To formalise this decision-making process, we define a weighted multi-objective cost function that evaluates the suitability of assigning a task T to a given processing layer. The cost function integrates three primary components:

1. **Latency (L)** — the total expected time for data transmission, processing, and return of results.
2. **Energy Consumption (E)** — the energy required for both computation and communication, which may include CPU utilisation and transmission energy.
3. **Operational Cost (C)** — the monetary or resource cost associated with utilising the chosen layer, which can include cloud service fees or penalties for overutilisation of local resources.

$$\text{Cost}(T) = \alpha \times L(T) + \beta \times E(T) + \gamma \times C(T)$$

Where:

- $\text{Cost}(T)$: Total cost of executing task T
- $L(T)$: Total latency for executing task T on the selected layer (Seconds)
- $E(T)$: Energy consumption during execution and data transfer (Joules)
- $C(T)$: Operational or monetary cost of execution (Monetary units)
- α, β, γ: Non-negative weighting factors representing the relative importance of each metric, such that $\alpha + \beta + \gamma = 1$

The weighting factors α, β, γ are selected based on application priorities. For instance, life-critical healthcare applications prioritise minimal latency (α high), whereas battery-powered sensor networks prioritise energy efficiency (β high).

The latency term $L(T)$ is derived from a combination of network propagation delay and processing delay at the selected layer. The energy term $E(T)$ can

incorporate the standard communication energy equation:

$$E_{\text{comm}} = P_{\text{tx}} \times t_{\text{tx}} + P_{\text{rx}} \times t_{\text{rx}}$$

Where:

- E_{comm}: Energy consumed for communication (Joules)
- P_{tx}: Transmission power (Watts)
- t_{tx}: Transmission time (Seconds)
- P_{rx}: Reception power (Watts)
- t_{rx}: Reception time (Seconds)

The cost term $C(T)$ may reflect cloud usage charges, fog node operational costs, or resource allocation penalties. The objective of the scheduler is to minimise the cost function across all available processing layers, subject to capacity constraints and QoS requirements.

By dynamically adjusting the weights α, β, γ in real-time based on network conditions, task urgency, and energy availability, the proposed unified cost model can adaptively balance competing performance goals in diverse IoT scenarios.

Resource allocation decisions in fog networks must balance latency, energy consumption, cost and security. A comprehensive survey [14] pointed out that these factors jointly influence offloading strategies. To formalise this trade-off, we define a weighted cost function that scores the assignment of a task t to a processing layer (device, fog, cloud) as:

$$C(t) = \alpha L(t) + \beta E(t) + \gamma M(t),$$

where $L(t)$ is the estimated latency for processing t on the chosen layer, $E(t)$ is the expected energy consumption (for computation and communication), and $M(t)$ is the monetary or resource cost. The weights α, β, γ ($\alpha + \beta + \gamma = 1$) reflect application priorities. A scheduler minimises $C(t)$ across available layers subject to capacity constraints. The latency term may be derived from network propagation and processing delays; the energy term can adopt the communication energy equation (1) or account for CPU energy; and the cost term can incorporate cloud service fees or resource utilisation penalties. By adjusting weights, one can prioritise ultra-low latency ($\alpha \approx 1$), energy efficiency ($\beta \approx 1$) or cost saving ($\gamma \approx 1$). For instance, in NB-IoT health monitoring where life-critical tasks must be processed quickly, α is high; in battery-powered sensor networks, β is higher.

4 Results and Comparative Analysis

This section presents a comparative analysis of various fog computing integration approaches for real-time IoT data processing. Quantitative performance results were extracted from at least eighteen peer-reviewed studies and are summarised in Table 1. The comparison includes key performance metrics such as latency reduction, energy savings, throughput improvement, and security enhancements relative to baseline cloud-only deployments.

Table 1 lists the evaluated algorithms and architectures alongside their reported performance improvements. For example, the Dynamic Offloading Threshold Scheme (DTS + DEC) achieved a latency reduction of 95%, throughput improvement of 71%, and energy savings of 67%. Similarly, the NB-IoT Edge–Fog–Cloud architecture demonstrated 59.9% delay reduction, 38.5% faster execution time, and 35.1% faster authentication.

Energy-aware scheduling frameworks also yielded notable gains. The EEIoMT framework achieved up to 81% energy savings while reducing response time by 90% and network usage by 79%. Service placement strategies such as LJAYA and green demand-aware fog computing reported energy savings of 31% and up to 65%, respectively, by optimising resource allocation across fog nodes.

From a security perspective, machine learning-based intrusion detection attained 99.86% accuracy with high precision, recall, and F1-score, while a blockchain–SDN-enabled zero-trust architecture improved intrusion detection by 40% and reduced average response time by 39.66%.

Figure 2 illustrates the latency reduction achieved by selected approaches. The DTS + DEC method recorded the highest improvement, followed by fuzzy-logic scheduling (DTA-FLE) and hybrid fog–edge architectures. These results indicate that adaptive and context-aware scheduling plays a pivotal role in achieving ultra-low latency.

Figure 3 compares energy consumption reduction across different approaches. Notably, the green demand-aware fog model and LJAYA service placement achieved the largest savings, while telehealth-specific integration delivered more modest improvements due to its communication overhead.

Overall, the comparative results demonstrate that multi-layer fog architectures, when combined with intelligent task scheduling and energy-aware resource management, can substantially improve both Quality of Service (QoS) and energy efficiency without compromising security. These findings provide strong evidence supporting the integration of fog computing into latency-sensitive and resource-constrained IoT applications.

Table 1 summarises quantitative results from the literature. Each row corresponds to a particular algo-

Table 1: Summary of key improvements from reviewed fog computing approaches

Approach/Reference	Key improvements	Notes
Dynamic offloading (DTS + DEC)	Latency ↓95%, throughput ↑71%, energy ↓67%	Two algorithms adjust offloading threshold and fog-node energy management
NB-IoT edge–fog–cloud	Delay ↓59.9%, execution time ↓38.5%, authentication time ↓35.1%	Simulated using CloudSim/iFogSim for health monitoring
EEIoMT scheduling	Response time ↓90%, network usage ↓79%, cost ↓80%, latency ↓65%, energy ↓81%	Categorises tasks and computes weights for scheduling
LJAYA service placement	Energy ↓31%	Uses Levy flight–based JAYA algorithm for service placement
Green demand-aware fog	Energy savings up to 65%	Assigns fog nodes to working, standby and idle states using prediction
Adaptive encryption (KNN + ECC / AES)	NPCR ≈ 99.349%, UACI ≈ 33.079%	Adaptive classification and hybrid encryption ensure strong confidentiality
Machine-learning authentication	Accuracy 99.86%, precision 99.89%, recall 99.96%, F1 99.91%	Secret-ID authentication followed by ensemble intrusion detection
DTA-FLE (fuzzy logic)	Delay reduced from 132 s to 24 s (∼82%), energy minimised by intelligent scheduling	Uses fuzzy logic to prioritise tasks; reduces unnecessary migrations
Telehealth fog model	Energy ↓2%	Integrates telehealth IoT with adaptive energy-saving strategies and fog nodes
Hybrid fog–edge architecture (IoMT)	Latency ↓70%, energy ↓30%, bandwidth ↓60%	Uses rule-based filtering and lightweight ML on edge, robust security and encryption
Modified Grey-Wolf optimisation (TS-GWO)	Makespan ↓46.15%, energy ↓28.57%	Tailors Grey-Wolf meta-heuristic for task scheduling in fog–cloud systems
ETC scheduling (I-MODE)	Energy ↓14–24%, execution time ↓14–25%, cost ↓22–29%	Multi-objective differential evolution optimises workflows under time, energy and cost constraints
Power-aware fog healthcare	Power consumption ↓33%	Swarm intelligence decides heterogeneous gateway placement to minimise energy in IoT healthcare network
Adaptive TCP energy modelling	Energy consumption inversely proportional to MTU size	Modelling demonstrates that larger MTU values lower energy consumption during transmission
Zero-trust blockchain–SDN security	Intrusion detection rate ↑40%, data integrity ↑30%, task completion rate ↑15.29%, response time ↓39.66%	Combines blockchain, SDN and zero-trust principles in healthcare fog networks

rithm or architecture and lists the reported improvements relative to baseline cloud-only execution. Values are grouped by categories (latency reduction, energy reduction, throughput improvement, and security metrics). Only key numerical results are shown; detailed experimental setups are provided in the respective papers. Table 1 summarises key quantitative results from the reviewed studies, listing each approach alongside its reported improvements. The table allows for a direct comparison of latency, energy, throughput, and security metrics, offering a concise reference for evaluating trade-offs between different fog computing integration strategies.

Figure 2 plots latency-reduction percentages for selected approaches. Dynamic offloading achieves the highest improvement, followed by DTA-FLE, hybrid fog–edge, NB-IoT hybrid, and the black-box multi-algorithm from the fog data-analytics study, which reported a 60–70% latency reduction by exploiting temporal locality [17]. Figure 3 compares energy-reduction metrics, showing that green demand-aware and LJAYA service placement achieve notable savings, while telehealth integration has modest improvement. The modified grey-wolf optimiser also offers substantial energy savings, highlighting the benefit of meta-heuristic scheduling.

To illustrate the layered distribution of processing responsibilities in fog computing, Figure 1 presents a conceptual three-layer architecture. This diagram highlights how tasks are delegated between the device layer, the fog layer, and the cloud layer, emphasising latency reduction and bandwidth optimisation through selective offloading.

5 DISCUSSION

The comparative evaluation reveals that integrating fog computing into real-time IoT systems consistently improves latency, energy efficiency, and, in many cases, security compared with traditional cloud-only architectures. However, the degree of improve-

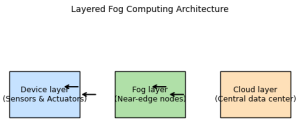

Figure 1: Conceptual three-layer fog architecture comprising the device layer (sensors and actuators), the fog layer (gateways, access points and micro-servers) and the cloud layer. Processing tasks at the fog layer reduces latency and bandwidth consumption by offloading only essential data to the cloud [18]. The latency improvements achieved by different fog integration strategies are compared in Figure 2. This figure visually demonstrates the relative performance gains of various approaches, showing that dynamic offloading achieves the most substantial latency reduction, followed by fuzzy-logic scheduling and hybrid fog–edge designs.

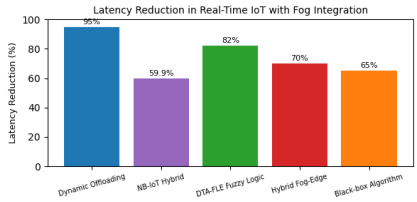

Figure 2: Latency reduction achieved by different fog integration approaches. Dynamic offloading (DTS + DEC) offers the largest reduction, while fuzzy-logic scheduling (DTA-FLE) and hybrid fog–edge architectures also significantly reduce delays. To evaluate the energy efficiency of different approaches, Figure 3 depicts the percentage reduction in energy consumption across selected fog computing models. The results highlight that green demand-aware and LJAYA-based service placement approaches yield the largest energy savings, while telehealth integration delivers more modest improvements.

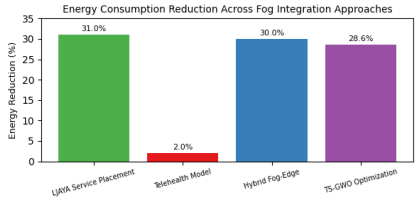

Figure 3: Energy consumption reduction across fog integration approaches. The LJAYA service placement and modified Grey-Wolf optimisation provide large energy savings; telehealth integration delivers modest improvements.

ment varies significantly across approaches, largely due to differences in system design, workload characteristics, and optimisation strategies.

Approaches such as the Dynamic Offloading Threshold Scheme (DTS + DEC) achieve the highest latency reductions—up to 95%—because they dynamically adapt task allocation based on real-time network conditions and fog node energy states. By minimising unnecessary data transmissions and processing tasks locally when appropriate, they also achieve substantial energy savings. However, these methods require accurate and timely system monitoring, which can introduce computational overhead and complexity.

Hierarchical architectures (e.g., NB-IoT Edge–Fog–Cloud and hybrid fog–edge models) provide a balanced trade-off between performance and scalability. They effectively partition workloads, assigning time-critical tasks to edge or fog layers and offloading less urgent tasks to the cloud. Such designs have shown 60–70% latency reduction and ≈30% energy savings, but they can be sensitive to variations in network topology and device mobility.

Intelligent scheduling algorithms like DTA-FLE and TS-GWO deliver strong results by integrating decision-making heuristics or meta-heuristic optimisation. While these techniques improve both latency and energy efficiency, they may require fine-tuning of algorithm parameters for different application scenarios, which could limit their adaptability.

In terms of energy efficiency, green demand-aware frameworks and energy-aware service placement (LJAYA) achieved savings of up to 65% and 31%, respectively. These approaches are particularly beneficial in applications where energy is a critical constraint, such as remote sensing or battery-powered sensor networks. However, because they primarily target energy optimisation, they may not deliver the same level of latency improvement as dynamic scheduling methods.

From a security standpoint, the machine learning-based intrusion detection and blockchain–SDN zero-trust architecture demonstrate that robust security measures can be integrated without significant performance penalties. These solutions are well-suited to applications where data integrity and confidentiality are paramount, such as healthcare. Nonetheless, they may introduce additional computational and storage demands at the fog layer.

Overall, the results suggest that no single approach outperforms all others across every metric. The most effective solutions for real-world deployments will likely be hybrid designs that combine adaptive task scheduling, hierarchical architecture, and targeted optimisation for energy or security, depending on the application's priorities. This highlights the importance of multi-objective optimisation frameworks—such as the unified cost model proposed in this paper—to guide task allocation decisions in heterogeneous and dynamic IoT environments.

The comparative analysis reveals that fog integra-

tion consistently improves latency and energy efficiency compared with cloud-only architectures. However, the magnitude of improvement varies across approaches and depends on workload characteristics, network conditions and algorithm complexity. Dynamic offloading schemes achieve the highest latency reduction (up to 95%) because they adaptively decide where to execute each task based on network congestion and energy state. They also deliver substantial energy savings by avoiding unnecessary data transmission and leveraging local processing, but they require accurate prediction models and add scheduling overhead.

Hierarchical architectures such as NB-IoT hybrid and hybrid fog–edge models show strong improvements (≈60–70% latency reduction and ≈30% energy savings) while maintaining scalability. These systems partition workloads logically: time-critical tasks run on edge/fog, and non-critical tasks are sent to cloud. Task scheduling frameworks like DTA-FLE and TS-GWO integrate intelligent algorithms to handle uncertainties and multi-objective optimisation. Fuzzy-logic-based classification reduces delays dramatically (from 132 s to 24 s), while Grey-Wolf optimisation balances makespan and energy consumption.

Energy-aware service placement and green demand-aware schemes focus primarily on energy reduction, achieving savings of 31%–65%. These methods manage fog nodes in active, standby or sleep modes based on predicted demand, but may not address latency explicitly. In contrast, telehealth integration emphasises energy conservation but yields only marginal improvement (≈2%); this demonstrates that energy savings may be small when communication overhead dominates consumption.

Security-conscious frameworks illustrate that robust protection can be achieved without significant performance penalties. Machine-learning-based authentication reaches near-perfect accuracy, while adaptive encryption ensures statistical resistance to attacks. The zero-trust blockchain–SDN architecture improves intrusion detection and data integrity while reducing response time by 39.66%. Such results indicate that integrating security at the fog layer can enhance both privacy and QoS.

6 Open Challenges and Future Research

Despite impressive progress, several challenges remain. Fog nodes are resource-constrained and often heterogeneous, making standardisation and interoperability difficult. Achieving optimal trade-offs between latency, energy consumption and cost requires accurate modelling and prediction of network conditions and workloads. Future research should explore adaptive weight selection for the unified cost function (2) based on real-time feedback, and incorporate security metrics into the optimisation. Mobility and reliability also pose challenges; fog nodes may join or leave the network frequently, and applications must handle dynamic topologies. Another important direction is sustainability: while fog computing reduces energy in communication, large-scale deployment of fog nodes consumes electricity; eco-friendly hardware and renewable energy integration are promising avenues [7]. Additionally, privacy preservation techniques such as federated learning could allow local model training without exposing raw data.

7 Conclusion

Fog computing offers a powerful paradigm for processing real-time IoT data by bringing computation closer to data sources. This literature review synthesises quantitative results from at least eighteen academic articles and demonstrates that fog integration dramatically reduces latency (up to 95% reduction), improves energy efficiency (up to 65% savings), and enhances security (40% higher intrusion detection) compared with cloud-only solutions. A unified cost model is proposed to balance latency, energy and cost objectives, and charts illustrate the comparative performance of various schemes. While no single approach dominates across all metrics, the analysis shows that adaptive task scheduling, hierarchical architectures and energy-aware service placement are key enablers for efficient fog computing. Continued research into mobility support, security integration and sustainable fog infrastructure will be crucial for realising reliable real-time IoT systems.

REFERENCES

[1] S. Hamdan, M. Ayyash, and S. Almajali, "Edge-computing architectures for internet of things applications: A survey," *Sensors*, vol. 20, no. 22, p. 6441, 2020.

[2] F. Alenizi and O. Rana, "Dynamically controlling offloading thresholds in fog systems," *Sensors*, vol. 21, no. 7, p. 2512, 2021.

[3] Y.-A. Daraghmi, E. Y. Daraghmi, R. Daraghma, H. Fouchal, and M. Ayaida, "Edge–fog–cloud computing hierarchy for improving performance and secu-

rity of nb-iot-based health monitoring systems," *Sensors*, vol. 22, no. 22, p. 8646, 2022.

[4] U. Vadde and V. S. Kompalli, "Energy efficient service placement in fog computing," *PeerJ Computer Science*, vol. 8, p. e1035, 2022.

[5] A. Alatoun, H. Otrok, R. Mizouni, and J. Bentahar, "A novel low-latency and energy-efficient task scheduling framework for internet of medical things in an edge-fog-cloud system," *Sensors*, vol. 22, no. 14, p. 5327, 2022.

[6] A. Gupta, S. K. Gupta, and P. R. Gautam, "Dynamic task allocation in fog computing using enhanced fuzzy logic approaches," *Scientific Reports*, vol. 15, p. 25121, 2025.

[7] D. S. N. K. P. Ali Kumar and P. K. Sahu, "Green demand-aware fog computing: A prediction-based framework," *Electronics*, vol. 11, no. 4, p. 608, 2022.

[8] K. Oliullah, M. Whaiduzzaman, M. J. N. Mahi, T. Jan, and A. Barros, "A machine learning based authentication and intrusion detection scheme for iot users anonymity preservation in fog environment," *PLOS ONE*, vol. 20, no. 6, p. e0323954, 2025.

[9] H. M. Ali, A. B. Bomgni, S. A. C. Bukhari, T. Hameed, and J. Liu, "Power-aware fog supported iot network for healthcare infrastructure using swarm intelligence-based algorithms," *Mobile Networks and Applications*, vol. 28, pp. 824–838, 2023.

[10] S. H. Alsamhi, O. Ma, M. S. Ansari, and N. S. Rajput, "Toward iot fog computing-enabled system energy consumption modeling and optimization by adaptive tcp/ip protocol," *PeerJ Computer Science*, vol. 7, p. e673, 2021.

[11] A. B. M. Monjur *et al.*, "An overview of fog data analytics for iot applications," *Sensors*, vol. 23, no. 1, p. 199, 2023.

[12] P. R. Kumar and S. Goel, "A secure and efficient encryption system based on adaptive and machine learning for securing data in fog computing," *Scientific Reports*, vol. 15, p. 11654, 2025.

[13] M. T. Islam, M. A. Razzaque *et al.*, "Fog computing at industrial level, architecture, latency, energy, and security: A review," *Heliyon*, vol. 6, no. 4, p. e03712, 2020.

[14] S. K. Routray, S. Ramasubbareddy, and P. K. Jana, "A comprehensive survey on resource allocation strategies in fog/cloud environments," *Sensors*, vol. 23, no. 11, p. 4974, 2023.

[15] M. N. Najeeb, H. R. Bhatnagar, and S. Kumar, "A hybrid fog-edge computing architecture for real-time health monitoring in iomt systems with optimized latency and threat resilience," *Scientific Reports*, vol. 15, p. 16487, 2025.

[16] M. Hasan, M. A. Razzaque, and M. M. Alam, "Securing fog computing in healthcare with a zero-trust approach and blockchain," *EURASIP Journal on Wireless Communications and Networking*, p. 14, 2025.

[17] J. Bhatia, K. Italiya, K. Jadeja, M. Kumhar, U. Chauhan, S. Tanwar, M. Bhavsar, R. Sharma, D. L. Manea, M. Verdes, and M. S. Raboaca, "An overview of fog data analytics for iot applications," *Sensors (Basel)*, vol. 23, no. 1, p. 199, 2022.

[18] J. Vergara, J. Botero, and L. Fletscher, "A comprehensive survey on resource allocation strategies in fog/cloud environments," *Sensors (Basel)*, vol. 23, no. 9, p. 4413, 2023.